

1 **Amendment to the Claims**

2 **In the Claims:**

3 Please cancel Claims 17 – 30 and amend Claims 1 and 3, as follows:

4 1. (Currently Amended) A programmable graphics pipeline for processing and rendering
5 variable length multimedia data, wherein different types of multimedia data are divided into
6 partitions of differing lengths, comprising:

7 (a) an instruction cache for storing at least graphics and media instructions;
8 (b) a first register file for storing the variable length multimedia data and
9 intermediate data;

10 (c) a first vector functional unit in communication with the instruction cache and
11 the register file, said vector functional unit performing at least graphics and media instructions, to
12 ~~produce~~ render at least graphics and media data; and

13 (d) an enhanced texture cache in communication with the first vector functional
14 unit, the first vector functional unit obtaining from said enhanced texture cache a vector of at least
15 one partition of the variable length multimedia data, wherein said enhanced texture cache is not
16 limited to storage of one type of multimedia data, but instead, is configured to stored multiple
17 partitions that are of differing lengths, to enable storage of the different types of multimedia data.

18 2. (Previously Canceled)

19 3. (Currently Amended) The programmable graphics pipeline of Claim 1, further comprising
20 a memory and an output buffer, wherein the enhanced texture cache comprises:

21 (a) a line buffer that provides multiple read ports for accessing the multimedia
22 data in multiple partitions of differing lengths; and

23 (b) a cache area in communication with the line buffer, said cache area storing the
24 multimedia data received from the memory and the output buffer.

25 4. (Original) The programmable graphics pipeline of Claim 3, further comprising an
26 enhanced texture address unit in communication with the enhanced texture cache, said enhanced
27 texture address unit being employed for converting multiple inverse-mapped source coordinates into
28 absolute memory addresses of the multimedia data in an arbitrary-sized block.

29 ///

30 ///

1 5. (Original) The programmable graphics pipeline of Claim 4, wherein the enhanced texture
2 address unit further generates filter coefficients for the multiple inverse-mapped source coordinates
3 for one of a bilinear filtering and a trilinear filtering of the data.

4 6. (Original) The programmable graphics pipeline of Claim 4, further comprising an
5 enhanced rasterization unit in communication with the enhanced texture address unit, said enhanced
6 rasterization unit being employed for generating:

- 7 (a) a plurality of destination coordinates for at least one of:
 - 8 (i) a plurality of primitives being rendered; and
 - 9 (ii) a media processing output; and
- 10 (b) a plurality of source coordinates for at least one of:
 - 11 (i) texture data associated with the plurality of destination coordinates for
12 the plurality of primitives; and
 - 13 (ii) media data associated with the plurality of destination coordinates for
14 the media processing.

15 7. (Original) The programmable graphics pipeline of Claim 6, wherein the plurality of
16 primitives include at least one of:

- 17 (a) dots;
- 18 (b) lines;
- 19 (c) triangles;
- 20 (d) rectangles; and
- 21 (e) polygons.

22 8. (Original) The programmable graphics pipeline of Claim 6, further comprising an
23 enhanced Z-buffer unit in communication with the enhanced rasterization unit and the enhanced
24 texture address unit, said enhanced Z-buffer unit being employed for loading source coordinates of a
25 primitive being rendered to provide the enhanced texture address unit with variable access to the
26 source coordinates if perspective address generation cannot be used.

27 9. (Original) The programmable graphics pipeline of Claim 8, wherein the enhanced
28 Z-buffer unit is further employed for determining a depth of a new source pixel in relation to an old
29 source pixel.

30 ///

1 10. (Original) The programmable graphics pipeline of Claim 1, further comprising a
2 blending unit that is in communication with the first vector functional unit, said blending unit being
3 employed for:

4 (a) combining a graphics data output from the first vector functional unit with a
5 color value to produce a blended value; and

6 (b) combining the blended value with a destination pixel value.

7 11. (Original) The programmable graphics pipeline of Claim 1, further comprising an output
8 buffer in communication with the first vector functional unit, said output buffer being employed for
9 concatenating one of successive image outputs and successive video outputs from the first vector
10 functional unit, to reduce a number of subsequent write transactions to a memory.

11 12. (Previously Presented) The programmable graphics pipeline of Claim 1, further
12 comprising an output buffer in communication with the first vector functional unit and the enhanced
13 texture cache, said output buffer being employed for:

14 (a) concatenating successive media processing output from the first vector
15 functional unit to produce concatenated data;

16 (b) providing the concatenated data as input to the enhanced texture cache; and

17 (c) converting a compressed format of destination coordinates into a plurality of
18 destination addresses for media processing output data.

19 13. (Original) The programmable graphics pipeline of Claim 10, further comprising a write
20 buffer in communication with the blending unit, said write buffer sending multiple write transactions
21 to a memory, to reduce page misses.

22 14. (Original) The programmable graphics pipeline of Claim 11, further comprising a write
23 buffer in communication with the output buffer, said write buffer sending multiple write transactions
24 to a memory, to reduce page misses.

25 15. (Original) The programmable graphics pipeline of Claim 1, further comprising a
26 configuration register in communication with the first vector functional unit, said configuration
27 register providing the first vector functional unit with partitioned data access parameters and a
28 location of a current instruction thereby instructing the vector functional unit to perform one of a
29 graphics process and a media process.

30 ///

1 16. (Original) The programmable graphics pipeline of Claim 1, further comprising:

2 (a) a second vector functional unit in communication with the instruction cache
3 and performing the graphics and media instructions being performed by the first vector functional
4 unit, to produce parallel graphics and media data; and

5 (b) a second register file in communication with the second vector functional unit,
6 said second register file storing additional multimedia data and additional intermediate data.

7 Claims 17 – 30 (Currently Canceled)